CIGRE-499
2022 CIGRE Canada Conference & Expo
Calgary, Alberta, Oct.31 – Nov. 3, 2022

TRANSMISSION STRUCTURES ADVANCED ANALYTICS AND MACHINE LEARNING ALGORITHMES

Ibrahim Hathout, Ph.D.
Hathout PGS Inc.
Mississauga, Ontario, Canada

Tariq Hathout, B. Eng.
SNC-Lavalin
Toronto, Ontario, Canada

SUMMARY

Most electric utilities use a proactive strategy to maintain existing transmission assets through regular inspection, engineering assessment, repair, and replacement programs. The goal is to move proactively to service existing assets before it results in costly failure, while at the same time not over-servicing assets that don't need it. Asset inspections provide critical information for improvement strategies and formulating transmission line reliability. The effectiveness of these strategies relies heavily on the quality of the mostly visual field inspection data. Because the procedure of assigning and combining rating information for line's components are based on subjective judgement and intuition, the resulting rating for structures with similar conditions can vary.

To reduce the cognitive type uncertainties inherent in visual inspection and condition assessment, the authors are proposing the use of machine learning algorithms to complement the visual inspection and human expert condition assessment using feed-forward artificial neural network (ANN) trained using a supervised backpropagation learning scheme. The ANN uses algorithms to minimize the errors and adjust the connection weights so that the behaviour of a well-trained ANN mimics the human brain in terms of processing incomplete and inaccurate data. The backpropagation technique employs a gradient descent algorithm to minimize the squared error between the network output values and the target values for these outputs. An activation function in a backpropagation network defines the way to obtain output of a neuron given the collective input from source ANN synapses. Because the backpropagation algorithm requires the activation function to be continuous and differentiable, the model uses a sigmoid (logistic) function to adjust the step, position, and mapping scope simultaneously, so it has stronger non-linear mapping capabilities.

The proposed model is a three-layer ANN trained to determine the remaining thickness of corroded steel members due to the two most dominant corrosion types: the surface and the pitting. The model scans the digital images of corroded steel structures to detect the type and severity of corrosion. For surface type corrosion, the model uses an RGB colour model to relate the colour of the rust to the severity of the corrosion. The model's training data consist of a selected collection of field measurements of rusted tower members. A digital picture of the rusted member is taken, then the rusted member is ground to near white, and the remaining thickness is measured. Collecting a large set of data for various corrosion

Ibrahim.hathout@hathoutpgs.com

colours and the corresponding thickness loss allows the trained model to accurately relate the colour of the rust to the thickness loss. The model is also capable of determining the severity and thickness losses due to pitting type corrosion. The model scans the digital image of the corroded steel member looking for pits. The model is trained to identify the pit's shape and colour and count the number of pits per 6.5 cm^2 (1 in^2). Once the number of pits is determined, the model maps the number of pits to ASTM G46 specification for pitting corrosion to predict the loss of metal due to pitting corrosion.

The model will considerably reduce the cognitive type uncertainties and bias inherent from visual inspection and assessment of existing transmission structures thus improving the efficiency of the repair, maintenance and refurbishment of existing transmission structures and ultimately improve the reliability and thus maximize the return on asset (ROA). A test case is presented to illustrate the power of the proposed model.

KEYWORDS

Machine Learning, Transmission Structures, Neural Networks, Damage Assessment, corroded steel

1. MACHINE LEARNING (ML)

ML is a subfield of artificial intelligence (AI) that provides computers with the ability to learn from data. It is the study of computer algorithms that can improve automatically through experience and by using data [1]. ML algorithms use training data to build models to make predictions or decisions without being explicitly programmed.

The ML algorithms are used in a wide variety of applications, such as in damage or condition assessment, image recognition, speech recognition, and computer vision, etc., where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks [2].

For ML programs to deliver expected results, access to high quality data is of paramount importance. Alberto Artasanchez [3] cited nine reasons for failure of ML programs such as lack of enough quality data, data bias, too much data, badly chosen programing tools and algorithms, etc. With the appropriate ML models, utilities can continually predict changes in their asset so that they are best able to predict what is next and where to invest and get maximum return on asset. As data is constantly added, the ML models ensure that the solution is continually updated, and prediction is more accurate.

2. CATEGORIES OF MACHINE LEARNING ALGORITHMS

There are two main categories of algorithms. These are:

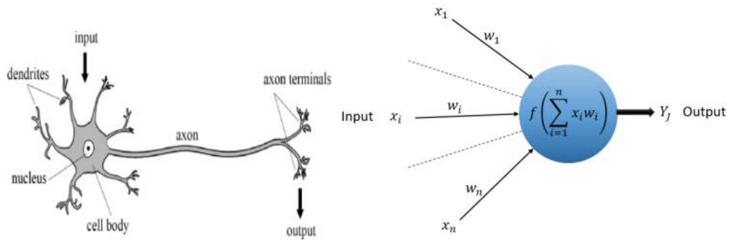
Supervised learning [4][5] is the most common type of machine learning. Supervised ML algorithms are designed to learn by example. When training a supervised learning algorithm, the training data will consist of inputs and the corresponding correct outputs (labelled). During training, the algorithm will search for patterns in the data that correlate with the desired outputs. After training, a supervised learning algorithm will take in new (unlabeled) inputs and will determine which label the new inputs will be classified as based on prior training data. The aim of a supervised learning model is to predict the correct label for newly presented input data. Artificial Neural Network is an example of supervised learning.

Unsupervised Learning - In this algorithm category, we do not have any target or output variable to predict or estimate. It is used for clustering populations in different groups, which is widely used for segmenting customers in different groups for specific intervention. Example of Unsupervised Learning is the K-means.

3. MULTILAYER PERCEPTRON (NEURAL NETWORKS):

One of the most powerful ML algorithms is the ANN. An ANN is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of many highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, can learn and be trained. An ANN is configured for a specific application, such as pattern recognition, image analysis, or data classification, through a learning process.

Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is also true for ANNs. An ANN incorporates the two fundamental components of biological neural nets: Neurons (nodes) and Synapses (weights) as shown in Fig. 1 (a). The neuron, Fig. 1 (b) receives the signals from the back neurons, sums the weighted signals and passes the sum to an activation function to obtain the output y_j. A typical 3-layer feed forward artificial neural network is displayed in Fig. 2.



(a) (b)

Fig. 1 - A biological neuron (a) in comparison to an artificial neuron (b)

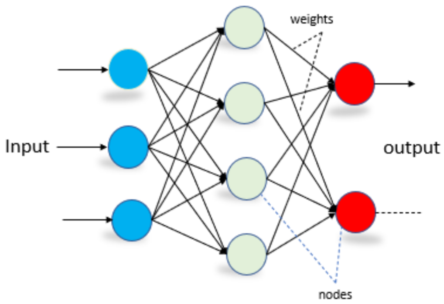


Fig. 2 – Typical 3-layer neural network

It should be noted that the ANNs take a different approach for solving problems than that of conventional algorithmic approach. The computer algorithm is a set of instructions designed to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem-solving capability of conventional computer algorithms to problems that we already understand and know how to solve. Computers would be much more useful if they could do things that we don't exactly understand or know how to do and be able to process incomplete, inaccurate, and fuzzy information like the human brain. It should be noted that the neural network uses algorithms to minimize the errors and adjust the connection weights. However, the behavior of a well-trained ANN will mimic the human brain in terms of processing incomplete and inaccurate data.

Activation function:

The activation function can adjust the step, position, and mapping scope simultaneously, so it has stronger non-linear mapping capabilities. Activation function in a back-propagation network defines the way to obtain output of a neuron given the collective input from source synapses. The back-propagation algorithm requires the activation function to be continuous and differentiable.

The most popular activation function used in neural networks is the sigmoid (logistic) function.

$$y = \frac{1}{1+e^{-x}} \tag{1}$$

This function is continuous and differentiable. The 1st derivative is given by:

$$\frac{dy}{dx} = \frac{e^{-x}}{(1+e^{-x})}\, y = y\,(1-y)$$

Fig. 3 shows how the sigmoid activation (squashing) function limits the node output. Assume that the sum of weighted signals to the node = 4.7. Apply the activation function $1/(1+e^{-4.7}) = 0.98$.

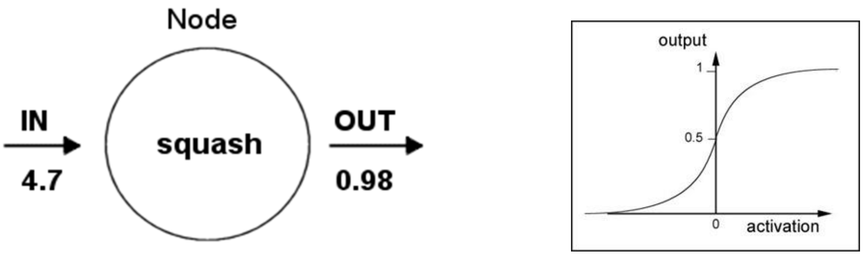


Fig. 3 – Squashing or activation function limits the node output.

To further control the node output, a constant or bias can be added to the node input. Bias in Neural Networks can be thought of as analogous to the role of a constant in a linear function, whereby the line is effectively transposed by the constant value.

Training the Artificial Neural Network:

Once a network has been structured for a particular application, that network is ready to be trained. ANNs are taught to adopt cognitive skills. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins.

Training is usually carried out by iterative updating of weights based on minimizing the mean square error. In the output layer, the error is the difference between the desired and the output values. Then the error is fed back through the steepest descent algorithm to update the weights of the network as shown in Fig. 4. The weights of the network are adjusted by the algorithm such that the error is decreased along a descent direction. Traditionally, two parameters, called learning rate, η, and momentum factor, ψ, are used for controlling the weight adjustment along the descent direction and for dampening error oscillations [6] [7].

The Back-propagation algorithm [8][9] can be described with the following three steps, which must be applied numerous times iteratively until the error becomes very small as shown in Fig. 4.

1. Forward computation of input signals of training samples and determination of neural network responses.
2. Computation of the error between desired response and neural network response.
3. Backward computation of the error and calculation of corrections to synaptic weights [8].

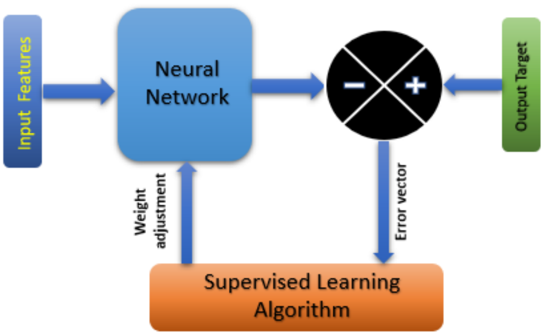


Fig. 4 – Training the neural network using Back-propagation algorithm

Calculating the weights in back-propagation learning algorithm:

Back-propagation algorithm employs gradient descent to attempt to minimize the squared error between the network output values and the target values for these outputs. The error E is defined as a sum of the squared errors over all the output unit "k" for all the training sets "d" and can be expressed as follows:

$$E(\vec{w}) \equiv \frac{1}{2} \sum_{d}^{D} \sum_{k}^{Output} (t_{kd} - O_{kd})^2 \tag{2}$$

Derivation of the back-propagation rule for one training set:

$$E_d(\vec{w}) \equiv \frac{1}{2} \sum_{k}^{outputs} (t_k - O_k)^2 \tag{3}$$

Gradient descent:

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \tag{4}$$

Where, η is the learning rate.

Using the chain-Rule; equation 5 can be written as follows:

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} x_{ji} \tag{5}$$

Similarly,

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial O_j} \frac{\partial O_j}{\partial net_j} \tag{6}$$

But,

$$net_j = \sum_i w_{ji} x_{ji} \tag{7}$$

and

$$\frac{\partial E_d}{\partial O_j} = \frac{1}{2} \frac{\partial}{\partial O_j} \sum_{k}^{outputs} (t_k - O_k)^2 = -(t_j - O_j) \tag{8}$$

$$\frac{\partial O_j}{\partial net_j} = \frac{\partial \sigma(net_j)}{\partial net_j} = O_j (1 - O_j) \tag{9}$$

Substitute, (6), (8), and (9) into equation (5), yield,

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} = \eta \left(t_j - O_j\right) O_j \left(1 - O_j\right) x_{ji} \tag{10}$$

$$w'_{ji} = w_{ji} + \Delta w_{ji} \tag{11}$$

The process is repeated until Δw_{ji} is minimized.

Where:

x_{ij}:	the i^{th} input to unit j
w_{ij}:	the weight associated with the i^{th} input to unit j
net_j:	the weighted sum of inputs for unit j
o_j:	the output computed by unit j
t_j:	the target output for unit j
σ:	the sigmoid function
outputs:	the set of units in the final layer of the network

Neural networks' early successes included predicting the stock market and self-driving car [10].

Unsupervised pre-training and increased computing power from the graphics processing unit, or GPU and distributed computing allowed the use of larger networks, particularly in image and visual recognition problems, which became known as "deep learning" [11].

4. USING ANN FOR CORROSION ANALYSIS OF TRANSMISSION STRUCTURES

The RGB color model [12]:

The RGB is an additive color model in which Red, Green, and Blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue. We use this model because the three wavelengths of the RGB primaries correspond to the signals that are transmitted from the eye to the brain. We see in RGB.

The main purpose of the RGB color model is for sensing, representation, and display of images in computers. For instance, if three colors are mixed at full strength, the combined RGB result is white and if all the components are at zero the result is black. Mixing red and green at full strength will result in Yellow, and so on. To recognize colors, first the image colors are divided into three categories:

Category A – Gray Scale:

Gray scale consists of black and gray regions. It was noticed that it is difficult to train neural networks to recognize different corrosion colors as well black and gray. It was therefore decided to train the network to discriminate between black and shades of gray first, and then re-train the network to recognize other colors.

Category B – Reddish:

Reddish consists of rust or corrosion colors: Brown, Red, Dark-Red, Orange, and combinations of the reddish shades.

Category C – Other:

All colors that don't have a dominant R component in the RGB scale are either green or blue shades. These cannot be part of the structures and represent the sky or vegetation background.

The selected part of the image is divided into small grids or pixels (default 10,000). Each pixel is a sample of an original image; more samples typically provide more accurate representations of the original. The intensity of each pixel is variable. In color image, a color is typically represented by the three component intensities Red, Green, and Blue (RGB). The color depth in an image is defined by the number of bits per pixel (bpp). For example, n-color bit = 2^n color.

Hathout et al. created a network that is trained to recognize corrosion types (surface, pitting) and the loss of thickness of steel members due to corrosion using neural network, color model, and shade removal algorithms [12]. The training data consist of large number of high-quality digital images of corroded steel and remaining thickness of selected steel members at various corrosion states.

Figure 5 show the digital image of a corroded tower scanned by Hathout et al. ANN model.

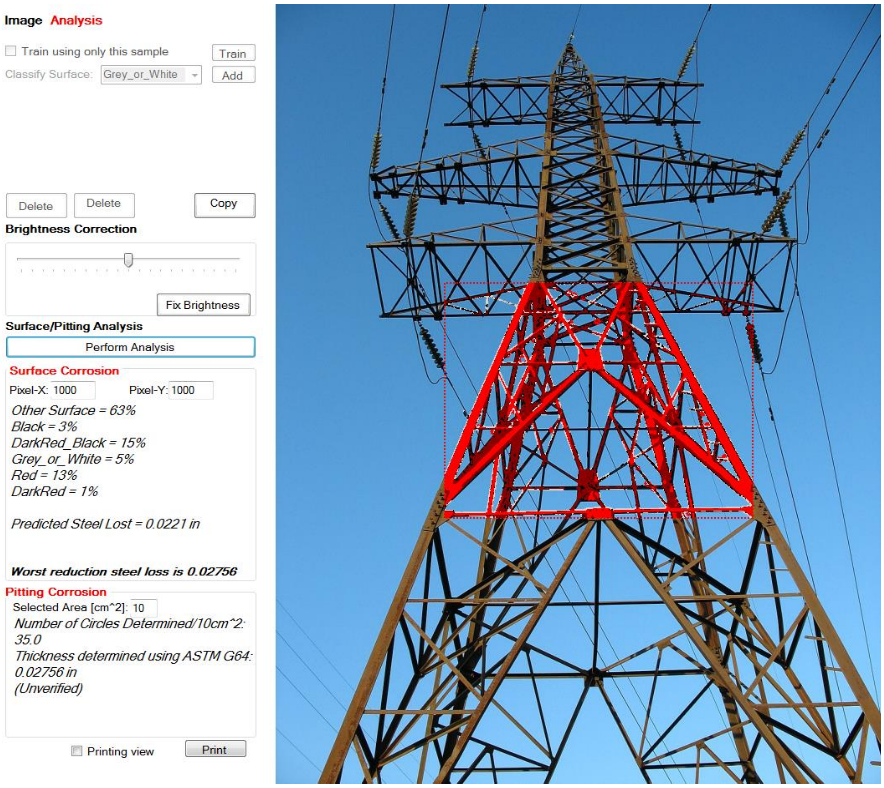


Fig. 5 - Image Analysis of selected area of the tower – The results are shown on the left

By means of the analysis results, the following observations can be made:

1. The expert system found a total of 35 pits and estimated the loss of thickness due to pitting corrosion to be 0.02756” (0.7mm) on average member thickness of 9.525mm or 7.3% thickness loss. The thickness loss due to surface corrosion is estimated to be 0.0221” (0.56mm) and the average thickness loss is 5.9%. Therefore, for analysis purpose a 7.3% thickness loss to be used.
2. 63% of the area was identified as other (Category C) or sky background. The expert system was able to recognize that the sky is not part of the steel.
3. ANN model was able to accurately predict the various colors in the image.

5. SHADE REMOVAL ALGORITHM

To improve the accuracy of ANN model, a shade removal algorithm for intrinsic image extraction is integrated into the ANN model. The Global Sparsity algorithm developed by Swedish researchers at Max Planck Institute [13] was tested on the image shown on the left in Fig. 6. The reflectance image is displayed on the right side of the same figure. The algorithm successfully removes the illumination effect from the original images and reveals the true color of the rust.

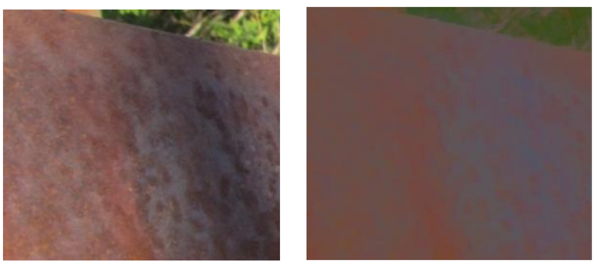


Fig. 6. Original corroded image (left) and the image’s reflectance derived from global sparsity (right)

6. CONCLUSIONS

Using the ANN model for determination of corrosion type and severity will greatly enhance the accuracy of damage assessment of corroded steel towers by reducing cognitive type uncertainty inherent in visual inspection and damage assessment.

The prediction of corrosion type and severity was very accurate when the image has no or minor illumination. Introducing the shade removal algorithm Global Sparsity enabled the model to extract the intrinsic (reflectance) images which are independent of illumination effects, thus revealing the images' true color information, thus increasing the accuracy of corrosion assessment regardless of the quality of the digital images.

BIBLIOGRAPHY

[1] Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892

[2] Hu, J.; Niu, H.; Carrasco, J.; Lennox, B.; Arvin, F., "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" IEEE Transactions on Vehicular Technology, 2020

[3] "9 Reasons why your machine learning project will fail". www.kdnuggets.com. Retrieved 2018-08-20.

[4] Stuart J. Russell, Peter Norvig (2010) Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall.

[5] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) Foundations of Machine Learning, The MIT Press.

[6] B. Gross, and D. Hanna, "Artificial neural networks capable of learning spatiotemporal chemical diffusion in the cortical brain" Pattern Recognition 43, pp. 3910-3921, 2010. © 2010 Elsevier Ltd.

[7] N. Nawi, R. Ransing and M. Ransing, "An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks", International Journal of Computational Intelligence Vol. 4 No. 1 2007.

[8] Z. Ismail and F. Jamaluddin, "A Backpropagation Method for Forecasting Electricity Load Demand". Journal of Applied Sciences 8(13): 2428-2434, 2008.

[9] James, et. al (2021), "An Introduction to Statistical Learning", Springer Texts Statistics, https://doi.org/10.10.1007/978-1-0716-1418-1_1.

[10] Domingos, Pedro (22 September 2015). The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World. chapter 4: Basic Books. ISBN 978-0465065707.

[11] Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press.

[12] Hathout, et al. (2017), "Digital Image Expert System for Corrosion Analysis of Steel Transmission Structures", 978-15386-2212-4/17, IEEE.

[13] Martin Kiefel, Lumin Zhang, Bernhard Scholkopf, Peter Vincent Gehler, and Carsten Rother. Recovering Intrinsic Images with a Global Sparsity Prior on Reflectance, Twenty-Fifth Annual Conference on Neural Information Processing Systems (NIPS), 2011